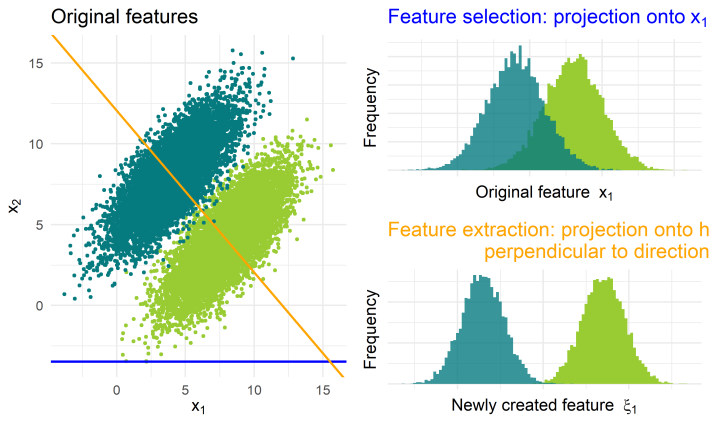
Supervised Learning

Feature Selection



Learning goals

- Understand that adding more features may be detrimental to prediction performance.
- Understand the benefits of keeping only useful features for the model.

INTRODUCTION

Feature selection deals with

- evaluating the influence of features on the model,
- techniques for choosing a suitable subset of features.

OVERVIEW

It is the task of statisticians, data analysts and machine learners to filter out the relevant information which is **useful** for prediction!

MOTIVATION

- The information about the target class is inherent in the features.
- Naive theoretical view:
  - More features
    - $\rightarrow$ more information
    - $\rightarrow$ more discriminant power
  - Model does not care about irrelevant features anyway (e.g. by estimating their coefficients to be 0).
- In practice there are many reasons why this is not the case!
- Moreover, optimizing is (usually) good, so we should optimize the input-coding.

MOTIVATION

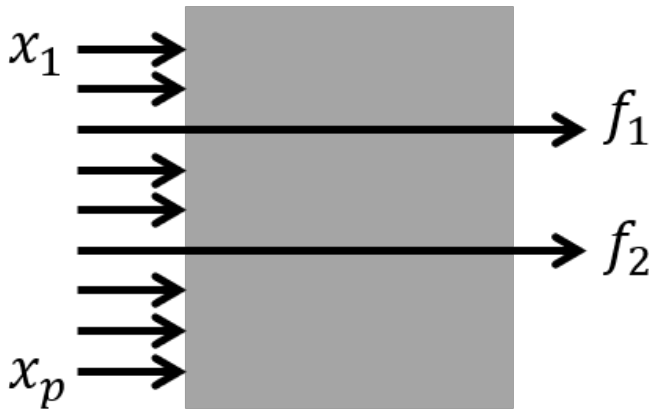
- In many domains we are confronted with an increasing number of features, many of which will be irrelevant or redundant, and multiple features of low quality.
- In domains with many features the underlying distribution function can be very complex and hard to estimate.
- Irrelevant and redundant features can “confuse” learners (recall the **curse of dimensionality**).
- Training data are limited.
- Computational resources are limited.
- Often the usual procedures are designed for $n > p$ problems.
- Thus, we either need
  - to adapt these procedures to high-dimensional data (e.g. by regularization),
  - design entirely new procedures,
  - or use the preprocessing methods addressed in this lecture.

SIZE OF DATASETS

- **Classical setting**: Up to around 10^2 features, feature selection might be relevant, but most of the time still manageable.
- **Datasets of medium to high dimensionality**: Around 10^2 to 10^3 features, basic methods still often work well.
- **High-dimensional data**: 10^3 to 10^7 features. Examples are e.g. micro-array / gene expression data and text categorization (bag-of-words features).
  If, in addition, observations are few, the scenario is called $p \gg n$.

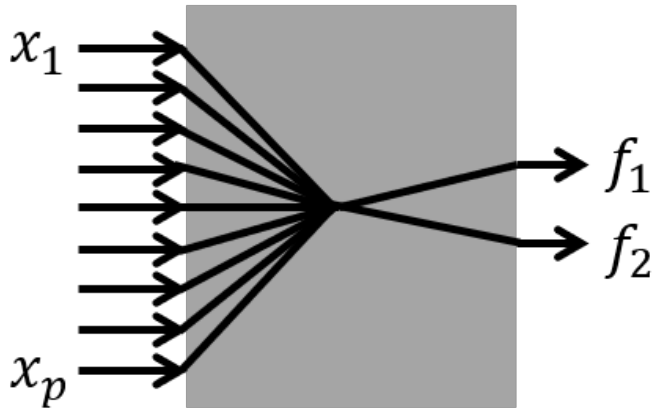
FEATURE SELECTION VS. EXTRACTION

Feature selection



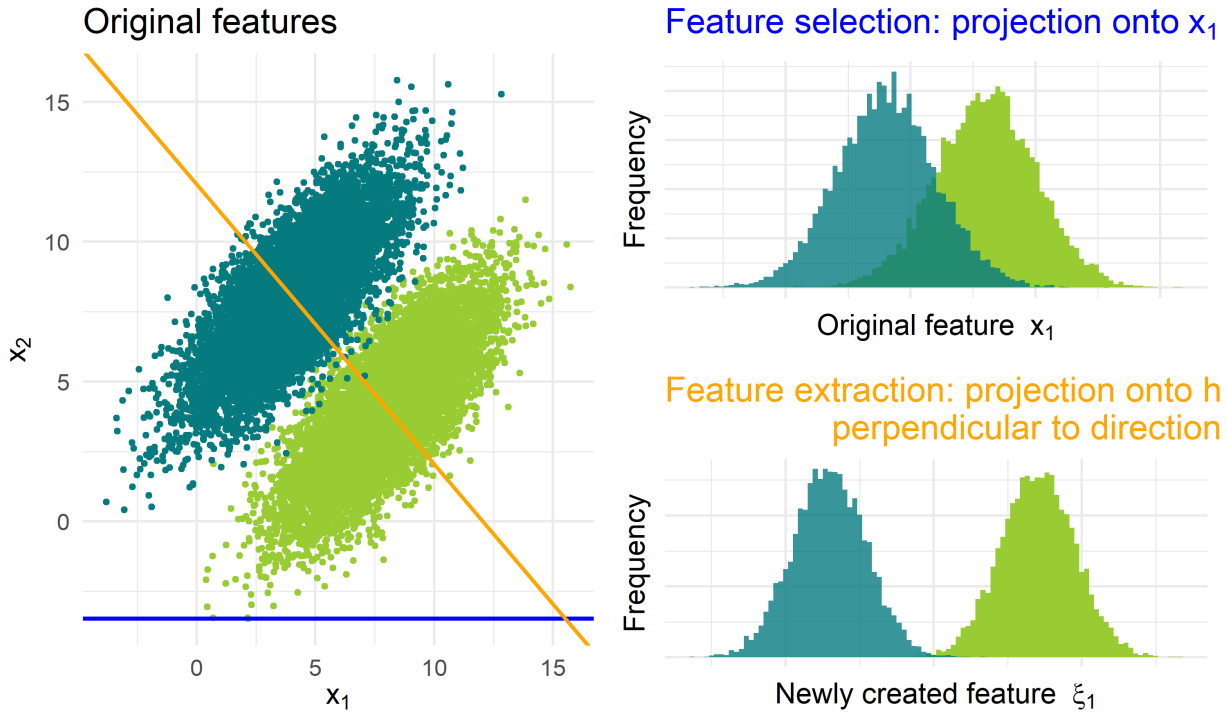
- Selects $\tilde{p} < p$ features.
- Creates a subset of the original features.
- Helps to understand the classification rules.

Feature extraction



- Maps inputs to $\tilde{p}$ new features.
- Forms linear and nonlinear combinations of the original features.

FEATURE SELECTION VS. EXTRACTION



Example for mixture of bivariate Gaussians. The projection onto the x_1 axis (i.e., feature selection) sees the two mixture components overlap, while projection onto the hyperplane perpendicular to the first principal component (i.e., feature extraction) separates the components.

FEATURE SELECTION VS. EXTRACTION

Both feature selection and feature extraction contribute to:

- Dimensionality reduction
- Simplicity of classification rules

Feature extraction can be supervised (Partial Least Squares (PLS), Sufficient Dimensional Reduction (SDR)) or unsupervised (Principal Component Analysis (PCA), Multidimensional Scaling (MDS), Manifold Learning).

OBJECTIVE FUNCTION

Given a set of features $\{1, \ldots, p\}$, the feature selection problem is to find a subset $S \subset \{1, \ldots p\}$ that maximizes the learner's ability to classify patterns. Formally, Ψ^* should maximize some objective function $\Psi : \Omega \to \mathbb{R}$:

$$\Psi^* = \arg\max_{S \in \Omega} \{\Psi(S)\}.$$

- Ω is the space of all possible feature subsets of $\{1, \ldots, p\}$.
- S can either be a subset of features (i.e., $\Omega \subseteq \mathcal{P}(\{1, \ldots, p\})$, $\mathcal{P}$ denoting the power set) or a bit vector (i.e., $\Omega = \{0, 1\}^p$) characterizing this subset. We will switch between those variants and the context will make clear which case we are referring to.
- Ψ can be different "things":
  - The BIC score in a linear regression model
  - The filter score of a Minimum Redundancy Maximum Relevance (mRMR) algorithm
  - The cross-validated test error of a learner

OBJECTIVE FUNCTION

How difficult is it to solve the introduced optimization problem, that is, to find the optimal feature set?

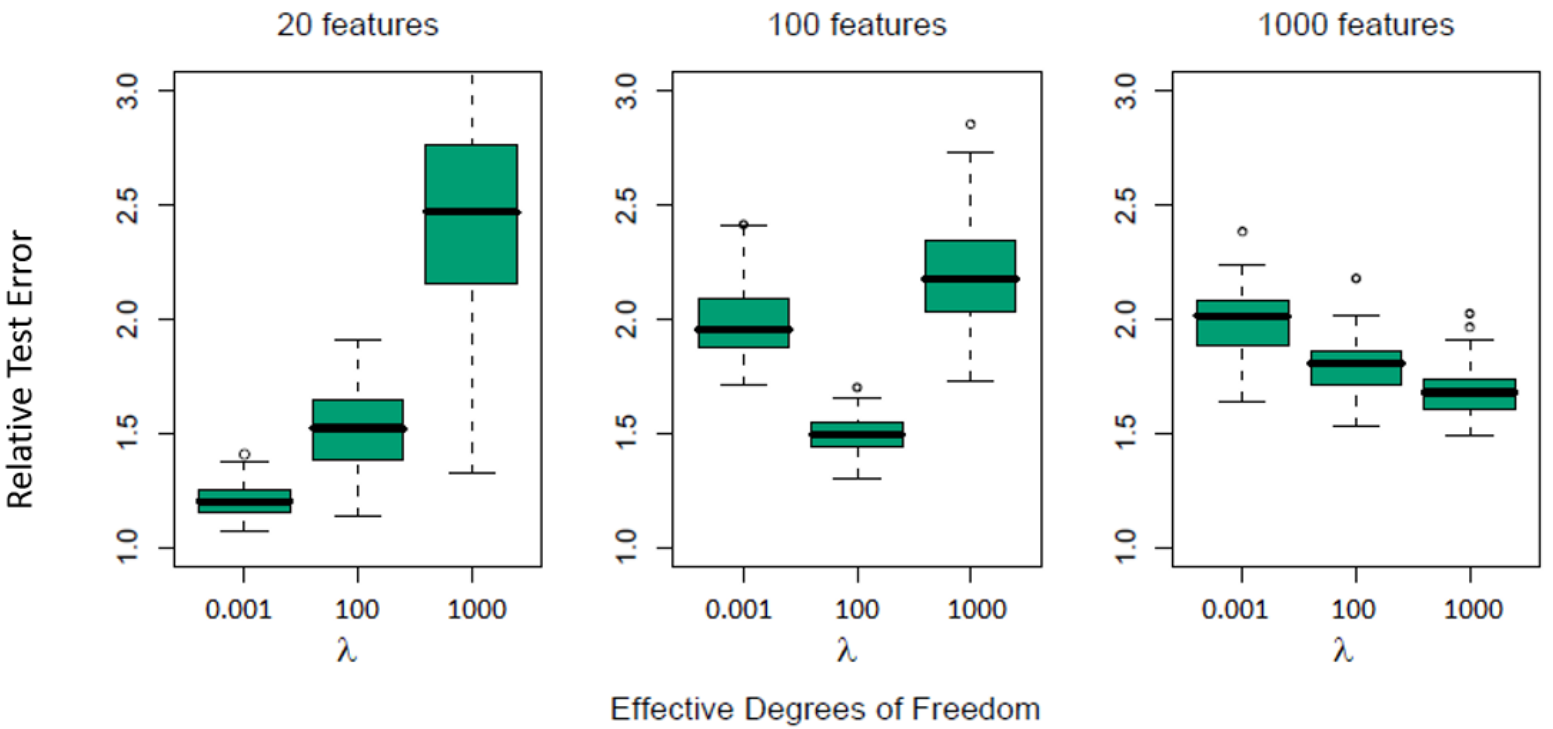
- The size of our search space (power set!) is 2^p.
- Hence, this is a discrete optimization problem.
- Of course this does not mean that we have to search the entire space, since there are more efficient search strategies.
- Unfortunately, for the general case, it can be shown that this problem will never be perfectly and efficiently solved (NP-hard).
- Thus our problem now consists of moving through the search space in a smart and efficient way, thereby finding a particularly good set of features.

MOTIVATING EXAMPLE: REGULARIZATION

- In case of $p \gg n$ “less fitting is better”.
- This can be demonstrated with the following simulation study.
- Investigation of three different dimensionalities of the input space: $p \in \{20, 100, 1000\}$.
- Data generation process for $n = 100$:
  - p random variables X are sampled from a standard Gaussian distribution with pairwise correlations of 0.2.
  - y is generated according to the linear model $y = \sum_{j=1}^{p} \mathbf{x}_j \theta_j + \sigma\epsilon$, where
    - ϵ and $\boldsymbol{\theta}$ are also sampled from standard Gaussian distributions, and
    - σ is chosen such that $\text{Var}(\mathbb{E}[y|X])/\sigma^2 = 2$.
  - 100 simulation runs are performed.
- A ridge regression model with $\lambda \in \{0.001, 100, 1000\}$ is fitted to the simulated data.

MOTIVATING EXAMPLE: REGULARIZATION

- Boxplots show the relative test error over 100 simulations for the different values of p and for the different regularization parameters $\lambda \in \{0.001, 100, 1000\}$ from left to right.
- Relative test error = test error / Bayes error σ^2.



MOTIVATING EXAMPLE: REGULARIZATION

- On the x-axis the effective degrees of freedom (averaged over the 100 simulation runs) are displayed:

$$df(\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

  where d are the singular values of x.
  - For $\lambda = 0$ (linear regression): $df(\lambda) = p$.
  - For $\lambda \to \infty$: $df(\lambda) \to 0$
- $\lambda = 0.001$ (20 df) has smallest relative rest error for $p = 20$.
- $\lambda = 100$ (35 df) has smallest relative rest error for $p = 100$.
- $\lambda = 1000$ (43 df) has smallest relative rest error for $p = 1000$.

$\Rightarrow$ More regularization for high dimensional data seems reasonable.

MOTIVATING EXAMPLE: DIFFERENT METHODS

Prediction results of eight different classification methods on micro-array data with $|\mathcal{D}_{\text{train}}| = 144$, $|\mathcal{D}_{\text{test}}| = 54$, $p = 16063$ genes and a categorical target which specifies the type of cancer out of 14 different cancer types.

Methods	CV errors (SE) Out of 144	Test errors Out of 54	Number of Genes Used
1. Nearest shrunken centroids	35 (5.0)	17	6,520
2. L_2-penalized discriminant analysis	25 (4.1)	12	16,063
3. Support vector classifier	26 (4.2)	14	16,063
4. Lasso regression (one vs all)	30.7 (1.8)	12.5	1,429
5. k-nearest neighbors	41 (4.6)	26	16,063
6. L_2-penalized multinomial	26 (4.2)	15	16,063
7. L_1-penalized multinomial	17 (2.8)	13	269
8. Elastic-net penalized multinomial	22 (3.7)	11.8	384

Hastie (2009). The Elements of Statistical Learning

MOTIVATING EXAMPLE: METHODS WITH INTEGRATED SELECTION

- We simulate data with function `sim.data` of the package **penalizedSVM**.
- With `sim.data` one can simulate micro-array data.
- Each simulated sample has `ng` genes.
- `nsg` genes are relevant and affect the class levels.
- The other `ng-nsg` have no influence.
- The gene ratios are drawn from a multivariate normal distribution.

MOTIVATING EXAMPLE: METHODS WITH INTEGRATED SELECTION

- Now we draw 200 observations with 50 features, 25 of which are positively and 25 are negatively correlated with the category.
- Furthermore, we create 50 irrelevant features.
- We compare several classification models regarding the misclassification rate.
- Since we have relatively few data, we use repeated cross-validation with 10 folds and 10 repetitions.

	rpart	lda	logreg	nBayes	7nn	rForest
all feat.	0.44	0.27	0.25	0.32	0.37	0.36
relevant feat.	0.44	0.18	0.19	0.27	0.33	0.30

The models with integrated selection do not work very well here!
If we knew the relevant features, we would achieve a significant improvement.

TYPES OF FEATURE SELECTION METHODS

In the following, we will get to know three different types of methods for feature selection:

- Filters
- Wrappers
- Embedded techniques

For each technique we will look at a simple example.

It should be noted that, in practice, complicated combinations of methods can also occur.